Stepper Motor Interfacing with Firebird V ATmega2560

Joel M. Pinto and Vishal Rajai
eYantra Summer Internship – 2014
Embedded Real-Time Systems Lab
Indian Institute of Technology, Bombay

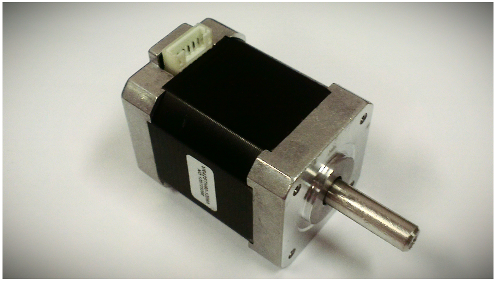
IIT Bombay
July 9, 2014

Agenda for Discussion

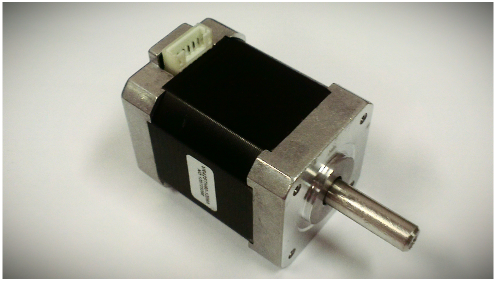
Prerequisite knowledge

1. Basic IO Interfacing using ports
2. Basic knowledge about timers in AVR

What is a stepper motor?

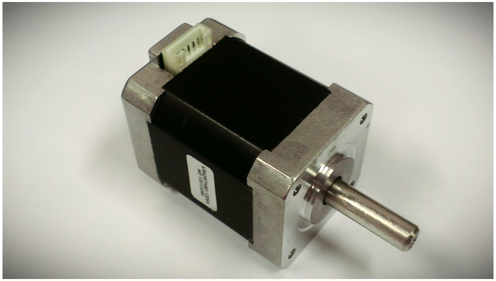


What is a stepper motor?



1. Rotates in discrete steps

What is a stepper motor?

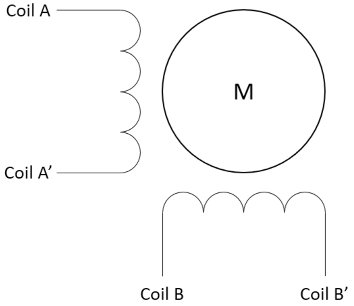


1. Rotates in discrete steps
2. Can hold or move to a position

Types of Stepper Motors

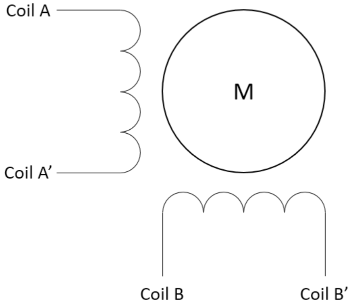
Types of Stepper Motors

- Bipolar
  - Has 4 wires

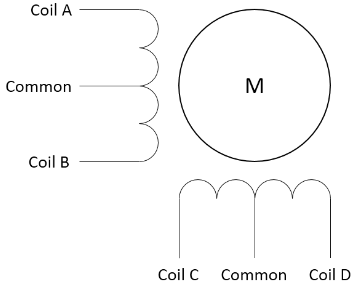


Types of Stepper Motors

- Bipolar
  - Has 4 wires

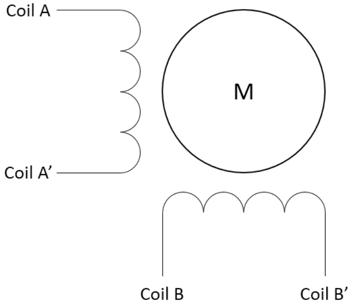


- Unipolar
  - Has 5 or 6 wires

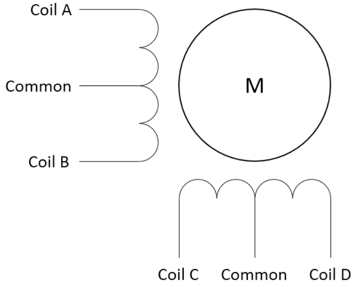


Types of Stepper Motors

- Bipolar
  - Has 4 wires



- Unipolar
  - Has 5 or 6 wires



We will use a unipolar stepper motor.

Stepping sequences

Stepping sequences

1. Wave Stepping

Stepping sequences

1. Wave Stepping
2. Full Stepping

Stepping sequences

1. Wave Stepping
2. Full Stepping
3. Half Stepping

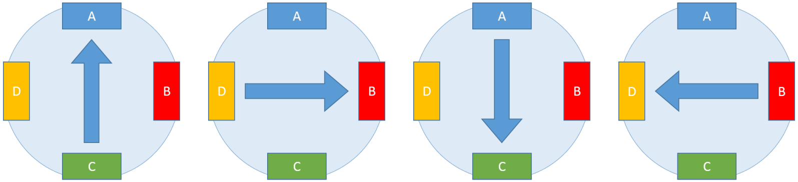
Wave Stepping

Wave Stepping

Step	**Coil A**	**Coil B**	**Coil C**	**Coil D**
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Table : Wave stepping sequence

Wave Stepping (contd.)



Stepper Motor's positions in the wave stepping sequence

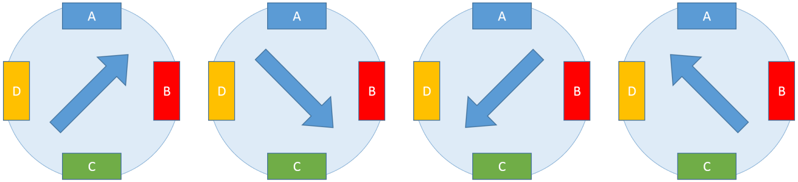
Full Stepping

Full Stepping

Step	Coil A	Coil B	Coil C	Coil D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Table : Full stepping sequence

Full Stepping (contd.)



Stepper Motor's positions in the full stepping sequence

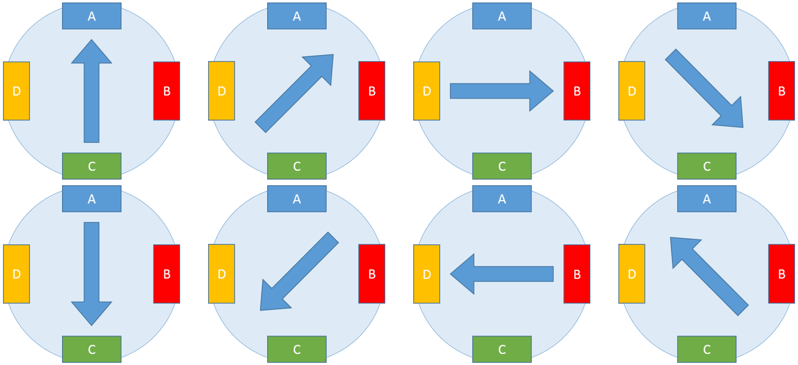
Half Stepping

Half Stepping

Step	Coil A	Coil B	Coil C	Coil D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Table : Half stepping sequence

Half Stepping (contd.)



Stepper Motor's positions in the half stepping sequence

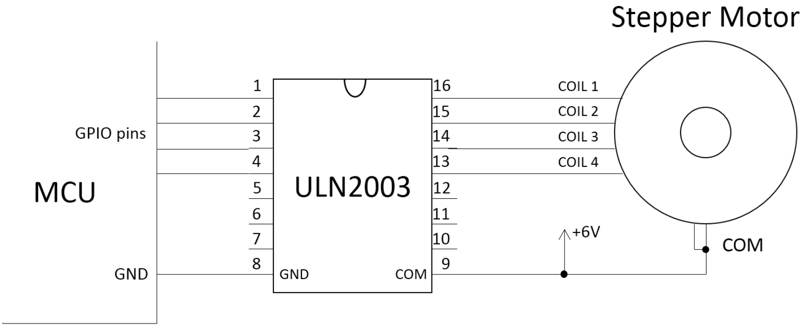
Comparison of stepping modes

Comparison of stepping modes

Stepping Mode	Wave Stepping	Full Stepping	Half Stepping
1 Torque	1 Lowest	1 Highest	1 Intermediate
2 Vibration	2 Intermediate	2 Highest	2 Lowest
3 Speed	3 Full	3 Full	3 Halved
4 Resolution	4 Normal	4 Normal	4 Doubled

Identifying the wires of a stepper motor

Stepper Motor Driver Circuit



Interfacing with ATmega2560
GPIO pins

Interfacing with ATmega2560
GPIO pins

Expansion Slot pin	MCU pin	Connected to
17	PL7	ULN2003 pin 1
18	PL6	ULN2003 pin 2
19	PD1	ULN2003 pin 3
20	PD0	ULN2003 pin 4
23	GND	ULN2003 pin 8

Table : GPIO pins used

Interfacing with ATmega2560

GPIO pins

Expansion Slot pin	MCU pin	Connected to
17	PL7	ULN2003 pin 1
18	PL6	ULN2003 pin 2
19	PD1	ULN2003 pin 3
20	PD0	ULN2003 pin 4
23	GND	ULN2003 pin 8

Table : GPIO pins used

1	4	...	17	20	21	24	...	53	56
2	3	...	18	19	22	23	...	54	55

Figure : Pin numbering on the expansion slot

Interfacing with ATmega2560
Timer Configuration

Interfacing with ATmega2560
Timer Configuration

✓ Time period of stepping = 3.333 ms ⇒ Frequency = 300 Hz

Interfacing with ATmega2560
Timer Configuration

✓ Time period of stepping = 3.333 ms ⇒ Frequency = 300 Hz

✓ 16-bit Timer1 in CTC mode ⇒ WGM13:0 = 4 (bin: 0100)

Interfacing with ATmega2560

Timer Configuration

- ✓ Time period of stepping = 3.333 ms ⇒ Frequency = 300 Hz
- ✓ 16-bit Timer1 in CTC mode ⇒ WGM13:0 = 4 (bin: 0100)
- ✓ Prescaler = 1

Interfacing with ATmega2560
Timer Configuration

✓ Time period of stepping = 3.333 ms ⇒ Frequency = 300 Hz

✓ 16-bit Timer1 in CTC mode ⇒ WGM13:0 = 4 (bin: 0100)

✓ Prescaler = 1

✓ Timer frequency = 300 Hz. So,

$$OCR1A = TOP = \frac{f_{CLK}}{f_{timer}} - 1 = \frac{14745600}{300} - 1 = 49151$$

Interfacing with ATmega2560
Timer Configuration

- ✓ Time period of stepping = 3.333 ms ⇒ Frequency = 300 Hz
- ✓ 16-bit Timer1 in CTC mode ⇒ WGM13:0 = 4 (bin: 0100)
- ✓ Prescaler = 1
- ✓ Timer frequency = 300 Hz. So,

$$OCR1A = TOP = \frac{f_{CLK}}{f_{timer}} - 1 = \frac{14745600}{300} - 1 = 49151$$

- ✓ Compare interrupt enabled

Interfacing with ATmega2560
Code

Interfacing with ATmega2560
Code

#include

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include "stepper.h"
```

Interfacing with ATmega2560
Code

#include

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include "stepper.h"
```

Interrupt Service Routine

```
ISR(TIMER1_COMPA_vect)
{
    wave_step(direction);
    stepcount++;
    if(stepcount > 200) //Change direction every revolution
    {
        direction *= -1;
        stepcount = 0;
    }
}
```

Interfacing with ATmega2560
Code (contd.)

Main Program

```
int main(void)
{
    stepper_port_init(); //Initialize ports
```

Interfacing with ATmega2560
Code (contd.)

Main Program

```
int main(void)
{
    stepper_port_init(); //Initialize ports

    cli(); //Clear global interrupts
    TCCR1B |= (1 << WGM12); //CTC mode (WGM13:0 = 0100)
    TIMSK1 |= (1 << OCIE1A); //Enable CTC interrupt
    sei(); //Enable global interrupts
```

Interfacing with ATmega2560

Code (contd.)

Main Program

```
int main(void)
{
    stepper_port_init(); //Initialize ports

    cli(); //Clear global interrupts
    TCCR1B |= (1 << WGM12); //CTC mode (WGM13:0 = 0100)
    TIMSK1 |= (1 << OCIE1A); //Enable CTC interrupt
    sei(); //Enable global interrupts

    OCR1A = (F_CPU / SPEED) - 1; //Set TOP
```

Interfacing with ATmega2560

Code (contd.)

Main Program

```
int main(void)
{
    stepper_port_init(); //Initialize ports

    cli(); //Clear global interrupts
    TCCR1B |= (1 << WGM12); //CTC mode (WGM13:0 = 0100)
    TIMSK1 |= (1 << OCIE1A); //Enable CTC interrupt
    sei(); //Enable global interrupts

    OCR1A = (F_CPU / SPEED) - 1; //Set TOP

    //Prescalar = 1
    TCCR1B |= ((0 << CS12) | (0 << CS11) | (1 << CS10));
```

Interfacing with ATmega2560

Code (contd.)

Main Program

```
int main(void)
{
    stepper_port_init(); //Initialize ports

    cli(); //Clear global interrupts
    TCCR1B |= (1 << WGM12); //CTC mode (WGM13:0 = 0100)
    TIMSK1 |= (1 << OCIE1A); //Enable CTC interrupt
    sei(); //Enable global interrupts

    OCR1A = (F_CPU / SPEED) - 1; //Set TOP

    //Prescalar = 1
    TCCR1B |= ((0 << CS12) | (0 << CS11) | (1 << CS10));

    while(1);
}
```

Thank You!

Send your queries to: helpdesk@e-yantra.org